Yiran Cui, Sebastian del Baño Rollin, Guido Germano

Full and fast calibration of the Heston stochastic volatility model

Article (Accepted version) (Refereed)

Original citation:
Cui, Yiran and del Baño Rollin, Sebastian and Germano, Guido (2017) *Full and fast calibration of the Heston stochastic volatility model.* European Journal of Operational Research, 263 (2). pp. 625-638. ISSN 0377-2217

DOI: 10.1016/j.ejor.2017.05.018

Full and fast calibration of the Heston stochastic volatility model

Yiran Cui[a,*], Sebastian del Baño Rollin[b], Guido Germano[a,c]

[a] *Financial Computing and Analytics Group, Department of Computer Science, University College London, Gower Street, London WC1E 6BT, United Kingdom*
[b] *School of Mathematical Sciences, Queen Mary University of London, Mile End Road, London E1 4NS, United Kingdom*
[c] *Systemic Risk Centre, London School of Economics and Political Science, Houghton Street, London WC2A 2AE, United Kingdom*

Abstract

This paper presents an algorithm for a complete and efficient calibration of the Heston stochastic volatility model. We express the calibration as a nonlinear least-squares problem. We exploit a suitable representation of the Heston characteristic function and modify it to avoid discontinuities caused by branch switchings of complex functions. Using this representation, we obtain the analytical gradient of the price of a vanilla option with respect to the model parameters, which is the key element of all variants of the objective function. The interdependence between the components of the gradient enables an efficient implementation which is around ten times faster than with a numerical gradient. We choose the Levenberg-Marquardt method to calibrate the model and do not observe multiple local minima reported in previous research. Two-dimensional sections show that the objective function is shaped as a narrow valley with a flat bottom. Our method is the fastest calibration of the Heston model developed so far and meets the speed requirement of practical trading.

Keywords: pricing, Heston model, model calibration, optimisation, Levenberg-Marquardt method.

1. Introduction

Pricing financial derivatives is an established problem in the operational research literature; see for example Fusai et al. [19] and references therein contained. Here we deal with the calibration of the Heston stochastic

*Corresponding author.
Email addresses: y.cui.12@ucl.ac.uk (Yiran Cui), s.delbanorollin@qmul.ac.uk (Sebastian del Baño Rollin), g.germano@ucl.ac.uk, g.germano@lse.ac.uk (Guido Germano)

volatility model [26], which is important and popular for derivatives pricing [5, 7, 38]. The particular topic of model calibration also involves numerical optimisation, which is a core subject of operational research.

A sophisticated model may reflect the reality better than a simple one, but usually is more challenging to implement and calibrate. This is especially true with mathematical models for the pricing of derivatives and the estimation of risk. The most basic model, introduced by Black and Scholes [8] (BS), assumes that the underlying price follows a geometric Brownian motion with constant drift and volatility. The price of a vanilla option is then given as a function of a single parameter, the volatility. However, the BS model does not adequately take into account essential characteristics of market dynamics such as fat tails and skewness of the distribution of log returns and the correlation between the value of the underlying and its volatility. It has also been observed that the volatility starts to fluctuate when the market reacts to new information. Thus, several extensions of the BS model were suggested thereafter, including the family of stochastic volatility (SV) models, which introduces a second Brownian motion to describe the fluctuation of the volatility. We study one of the most important SV models; it was proposed by Heston [26] and is defined by the system of stochastic differential equations

$$\mathrm{d}S_t = \mu S_t \mathrm{d}t + \sqrt{v_t} S_t \mathrm{d}W_t^{(1)}, \tag{1a}$$

$$\mathrm{d}v_t = \kappa(\bar{v} - v_t)\mathrm{d}t + \sigma\sqrt{v_t}\mathrm{d}W_t^{(2)}, \tag{1b}$$

with

$$\mathrm{d}W_t^{(1)}\mathrm{d}W_t^{(2)} = \rho \mathrm{d}t, \tag{1c}$$

where S_t is the underlying price and v_t its variance; the parameters $\kappa, \bar{v}, \sigma, \rho$ are respectively called the mean-reversion rate, the long-term variance, the volatility of volatility, and the correlation between the Brownian motions $W_t^{(1)}$ and $W_t^{(2)}$ that drive the underlying and its variance; moreover there is a fifth parameter v_0, the initial value of the variance.

Model calibration is as crucial as the model itself. Calibration consists in determining the parameter values so that the model reproduces market prices as accurately as possible. Both the accuracy and the speed of calibration are important because practitioners use the calibrated parameters to price a large number of complicated derivative contracts and to develop high-frequency trading strategies.

In this paper, we propose to efficiently calibrate the Heston model using an analytical gradient and numerical optimisation. In Section 2, we briefly review the existing research. In Section 3, we formulate the objective function and derive its analytical gradient. In Section 4, we present a complete algorithm to calibrate the Heston model using the Levenberg-Marquardt (LM) method. We also discuss some points where a carefully designed nu-

merical scheme may improve the performance. In Section 5, we present numerical results.

Throughout, we use bold uppercase letters for matrices, e.g. $\boldsymbol{J}$, and bold lowercase letters for column vectors, e.g. $\boldsymbol{\theta}$; a superscript $^\intercal$ for the transpose of a matrix or vector; $\boldsymbol{e}$ for a vector of all ones $[1, \ldots, 1]^\intercal$; $\mathrm{E}[\cdot]$ for expectations; $\mathbf{1}_A(\cdot)$ for the indicator function of the set A; $\mathrm{Re}(\cdot)$ for the real part and $\mathrm{Im}(\cdot)$ for the imaginary part of a complex number; $\|\cdot\|$ for the l_2-norm; $\|\cdot\|_\infty$ for the l_∞-norm; log for the natural logarithm.

2. Previous work on the calibration of the Heston model

In the literature, there are two main approaches to calibrate the Heston model: *historical* and *implied*. The first fits historical time series of the prices of an option with a fixed strike and maturity, typically by the maximum likelihood method or the efficient method of moments [1, 17, 27]. The second fits the volatility surface of an underlying at a fixed time, i.e., options with several strikes and maturities, to obtain the implied parameter set. Our work follows the second approach, as that is what is used in real-time pricing and risk management. In the following, we survey obstacles and existing methods for the Heston model calibration related to the second approach.

2.1. Recognised difficulties

Firstly, the calibration is in a five-dimensional space. There is no consensus among researchers on whether the objective function for the Heston model calibration is convex or irregular. The results of some proposed methods [10, 23, 34] depend on the initial point, which was attributed to a non-convex objective function, but might simply reflect on the inadequacy of the methods. To find a reasonable initial guess, short-term or long-term asymptotic rules are used; see Jacquier and Martini [28] for a detailed review. However, recently Gerlich et al. [22] claimed a convergence to the unique solution independent of the initial guess and suggested that the Heston calibration problem may have some inherent structure leading to a single stationary point. On the other hand, dependencies among the parameters do exist. For example, it is known that κ and σ offset each other: $\lim_{t\to+\infty} \mathrm{Var}(v_t) = \sigma^2\bar{v}/\kappa$, so that a parameter set with large values of κ and σ gives a fit comparable to a set with small values of κ and σ. Intuitively, the fact that different parameter combinations yield similar values of the objective function can be due to the objective function being flat close to the optimum; see Section 5 in this paper.

Secondly, the analytical gradient for the Heston calibration problem is hard to find and has not been available so far because it was believed that the expression of the Heston characteristic function is overly complicated to

provide an insightful analytical gradient: of course, a gradient can be obtained with symbolic algebra packages, but the resulting expressions are intractable. Instead, numerical gradients obtained by finite difference methods have been used in gradient-based optimisation methods; however, numerical gradients have a larger computational cost and a lower accuracy.

2.2. Existing methods

We review some heuristics to reduce the dimension of the calibration and then the optimisation methods that have been applied so far.

2.2.1. Heuristics for dimension reduction

Since the Heston parameters are closely related to the shape of the implied volatility surface [13, 21, 23, 29] (v_0 controls the position of the volatility smile, ρ the skewness, κ and σ the convexity, and κ times the difference between v_0 and $\bar{v}$ the term structure of implied volatility), efforts have been made to simplify the calibration to a lower dimension by presuming some of the parameter values based on knowledge available for the specific volatility surface. The initial variance v_0 is usually set to the short-term at-the-money (ATM) BS implied variance, which is based on the term structure of the BS implied volatility in the Heston model [21, p. 34-35]. A practical calibration experiment [10, p. 29-30] verified the linearity between the initial variance and the BS implied variance for maturities in the range of 1 to 2 months. Clark [13, Eq. (7.3)] suggested the heuristic assumption $\kappa = 2.75/\tau$ and $\bar{v} = \sigma_{\mathrm{ATM}}(\tau)$, where $\sigma_{\mathrm{ATM}}(\tau)$ is the ATM implied volatility with time to maturity τ. Chen [10] proposed a fast intraday recalibration by fixing κ to the same as yesterday's and v_0 to the 2-month ATM implied volatility, which are heuristics actually adopted in the industry. These assumptions help with an incomplete calibration, but may misguide the iterate to a limited domain and thus to a wrong convergence point.

2.2.2. Stochastic optimisation methods

Researchers who believed that a descent direction is unavailable have devoted their attention to stochastic optimisation methods, including Wang-Landau [10], differential evolution and particle swarm [24], simulated annealing [35], etc. To increase the robustness, a deterministic search such as Nelder and Mead using the MATLAB function `fminsearch` is often combined with these stochastic optimisation algorithms. Almost all research using stochastic techniques reports issues with performance. GPU technology has been applied with simulated annealing to speed up the calibration of the SABR model, a member of the family of SV models. Using 2 nVIDIA Geforce GTX470 GPUs it took 421.72 seconds to calibrate 12 instruments achieving a 10^{-2} maximum error [18], which is still too slow for real-time use.

2.2.3. Deterministic optimisation methods

Deterministic optimisation solvers available with commercial packages have been proved to be unstable as the performance largely depends on the quality of the initial guess: this applies to the Excel built-in solver [34] and to the MATLAB solver `lsqnonlin` [6, 20, 35]. Gerlich et al. [22] adopted a Gauss-Newton framework and kept the feasibility of the iterates by projecting to a cone determined by the constraints. The gradient of the objective function was calculated by finite differences and thus costs a large number of function evaluations.

To sum up, existing calibration algorithms are either based on ad hoc assumptions or not fast or stable enough for practical use. In this work, we will focus on deterministic optimisation methods without any presumption on the values of the parameters.

3. Problem formulation and gradient calculation

The idea of calibrating a volatility model is to minimise the difference between the vanilla option price calculated with the model and the one observed in the market. In this section, we first formulate the calibration problem in a least-squares form. Then, we present the pricing formula of a vanilla option under the Heston model with four algebraically equivalent representations of the characteristic function, discussing their numerical stability and suitability for analytical derivation. We calculate the analytical gradient of the objective function which can be used in any gradient-based optimisation algorithm.

3.1. The inverse problem formulation

Denote by $C^*(K_i, \tau_i)$ the market price of a vanilla call option with strike K_i and time to maturity $\tau_i := T_i - t$, $C(\boldsymbol{\theta}; K_i, \tau_i)$ the price computed via the Heston analytical formula (9) with the parameter vector $\boldsymbol{\theta} := [v_0, \bar{v}, \rho, \kappa, \sigma]^\intercal$. We assemble the residuals for the n options to be calibrated

$$r_i(\boldsymbol{\theta}) := C(\boldsymbol{\theta}; K_i, \tau_i) - C^*(K_i, \tau_i), \qquad i = 1, \ldots, n \tag{2}$$

in the residual vector $\boldsymbol{r}(\boldsymbol{\theta}) \in \mathbb{R}^n$, i.e.,

$$\boldsymbol{r}(\boldsymbol{\theta}) := [r_1(\boldsymbol{\theta}), r_2(\boldsymbol{\theta}), \ldots, r_n(\boldsymbol{\theta})]^\intercal . \tag{3}$$

We treat the calibration of the Heston model as an inverse problem in the nonlinear least-squares form

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^m} f(\boldsymbol{\theta}), \tag{4}$$

where $m = 5$ indicates the dimension, and

$$f(\boldsymbol{\theta}) := \frac{1}{2} \|\boldsymbol{r}(\boldsymbol{\theta})\|^2 = \frac{1}{2} \boldsymbol{r}^\intercal(\boldsymbol{\theta}) \boldsymbol{r}(\boldsymbol{\theta}). \tag{5}$$

Since there are many more market observations than parameters to be found, i.e., $n \gg m$, the calibration problem is overdetermined.

Our objective function is the sum of squared price differences on a set of strikes and maturities. The precise choice of the objective function is not trivial and, in an accounting sense, ultimately depends on the trade population in an actual trading book. The choice of strikes and maturities is also not trivial. For example, choosing the same strikes (even as a percentage of spot [11, Table 1]) will make these strikes more out of the money (OTM) for shorter maturities; this could lead to miscalibration of the longer-dated smile and to computational underflow for the shorter-ended options. To remedy this, we adopted the standard approach in foreign exchange (FX) which uses strikes defined by given values of BS delta (the sensitivity of the BS option price with respect to the spot price of the underlying), namely 10% and 25% delta call and put options as well as the ATM strike [13, Section 3.3]; see Section 5.

With regard to the actual objective function, other possibilities have been considered and can lead to different results [12]. One such function used in industry is the sum of squared differences of implied volatilities; this quantity can be approximated dividing the price difference by BS vega (the sensitivity of the BS option price with respect to the implied volatility)[1] and is relevant when the price bid/offer spread in volatility terms is independent of strike [14, Section 13.2]. This is not always the case, as less liquid OTM options are quoted with a wider bid/offer spread in volatilities. Calibrating to implied volatility will cause OTM options, with lower vega, to weigh more in the objective function, and so the resulting calibration, compared to the one based on price, will privilege OTM over ATM options. As our algorithm also approximates vega, future work might explore the optimisation problem in terms of implied volatility. However, since we derive the gradient from the pricing formula, here it is most straightforward and consistent to minimise the pricing error.

Before applying any technique to solve the problem (4) and (5), one needs to bear in mind that the evaluation of $C(\boldsymbol{\theta}; K_i, \tau_i)$ is expensive for the purpose of calibration; hence, one would like to minimise the number of computations of Eq. (9) when designing the algorithm. Moreover, the explicit gradient of $C(\boldsymbol{\theta}; K_i, \tau_i)$ with respect to $\boldsymbol{\theta}$ is not available in the literature as it is deemed to be overly complicated. This is indeed true if one

[1] BS vega is the quantity relevant for calibration to implied volatilities. There are other notions of vega. "Model vega" is the sensitivity of the option price in a particular model with respect to volatility. "Trader vega", used in hedging positions, is the result of bumping actual market prices prior to calibration and pricing. In FX, the market practice is to quote OTM options as risk-reversal and strangle volatilities; sensitivities to these quantities are often called rega and sega. The hope is that all these different measures are roughly comparable.

starts from the commonly used expressions for the characteristic function by Heston [26, Eq. (17)] or Schoutens et al. [39, Eq. (17)]. However, as shown in the next section, a more convenient choice of the functional form of the characteristic function by del Baño Rollin et al. [15, Eq. (6)] eases the derivation of its analytical gradient.

3.2. Pricing formula of a vanilla option and representations of the characteristic function

For a spot price S_t, an interest rate r and a dividend rate q, the price of a vanilla call option with strike K and time to maturity $\tau := T - t$ is

$$\begin{aligned} C(\boldsymbol{\theta}; K, \tau) &= e^{-r\tau}\mathrm{E}[(S_T - K)\mathbf{1}_{\{S_T \geq K\}}(S_T)] && \text{(6a)} \\ &= e^{-r\tau}\left(\mathrm{E}[S_T\mathbf{1}_{\{S_T \geq K\}}(S_T)] - K\mathrm{E}[\mathbf{1}_{\{S_T \geq K\}}(S_T)]\right) && \text{(6b)} \\ &= S_t e^{-q\tau} P_1(\boldsymbol{\theta}; K, \tau) - K e^{-r\tau} P_2(\boldsymbol{\theta}; K, \tau). && \text{(6c)} \end{aligned}$$

In the Heston model, $P_1(\boldsymbol{\theta}; K, \tau)$ and $P_2(\boldsymbol{\theta}; K, \tau)$ are solutions to certain pricing PDEs [26, Eq. (12)] and are given as

$$P_1(\boldsymbol{\theta}; K, \tau) = \frac{1}{2} + \frac{1}{\pi}\int_0^\infty \mathrm{Re}\left(\frac{e^{-iu\log\frac{K}{S_0}}}{iu}\frac{\phi(\boldsymbol{\theta}; u - i, \tau)}{\phi(\boldsymbol{\theta}; -i, \tau)}\right)\mathrm{d}u, \tag{7}$$

$$P_2(\boldsymbol{\theta}; K, \tau) = \frac{1}{2} + \frac{1}{\pi}\int_0^\infty \mathrm{Re}\left(\frac{e^{-iu\log\frac{K}{S_0}}}{iu}\phi(\boldsymbol{\theta}; u, \tau)\right)\mathrm{d}u, \tag{8}$$

where i is the imaginary unit, $\phi(\boldsymbol{\theta}; u, \tau)$ is the characteristic function of the logarithm of the stock price process, $\phi(\boldsymbol{\theta}; -i, \tau) = F/S_0$, and $F := S_t e^{(r-q)\tau}$ is the forward price. Thus, the formula for pricing a vanilla call option becomes

$$\begin{aligned} C(\boldsymbol{\theta}; K, \tau) = \frac{1}{2}(S_t e^{-q\tau} - K e^{-r\tau}) \\ + \frac{e^{-r\tau}}{\pi}\left[S_0\int_0^\infty \mathrm{Re}\left(\frac{e^{-iu\log\frac{K}{S_0}}}{iu}\phi(\boldsymbol{\theta}; u - i, \tau)\right)\mathrm{d}u\right. \\ \left. - K\int_0^\infty \mathrm{Re}\left(\frac{e^{-iu\log\frac{K}{S_0}}}{iu}\phi(\boldsymbol{\theta}; u, \tau)\right)\mathrm{d}u\right]. \end{aligned} \tag{9}$$

The characteristic function was originally given by Heston [26, Eq. (17)] as

$$\begin{aligned} \phi(\boldsymbol{\theta}; u, \tau) := \mathrm{E}\left[\exp\left(iu\log\frac{S_t}{S_0}\right)\right] = \exp\left\{iu\log\frac{F}{S_0}\right. \\ \left. + \frac{\kappa\bar{v}}{\sigma^2}\left[(\xi + d)\tau - 2\log\frac{1 - g_1 e^{d\tau}}{1 - g_1}\right] + \frac{v_0}{\sigma^2}(\xi + d)\frac{1 - e^{d\tau}}{1 - g_1 e^{d\tau}}\right\}, \end{aligned} \tag{10}$$

where

$$\xi := \kappa - \sigma\rho i u, \tag{11a}$$

$$d := \sqrt{\xi^2 + \sigma^2(u^2 + iu)}, \tag{11b}$$

$$g_1 := \frac{\xi + d}{\xi - d}. \tag{11c}$$

Kahl and Jäckel [30] pointed out that when evaluating this form as a function of u for moderate to long maturities, discontinuities appear because of the branch switching of the complex power function $G^\alpha(u) = \exp(\alpha \log G(u))$ with $G(u) := (1 - g_1 e^{d\tau})/(1 - g_1)$ and $\alpha := \kappa\bar{v}/\sigma^2$, which appears in Eq. (10) as a multivalued complex logarithm. This depends on the fact that $G(u)$ has the shape of a spiral as u increases, and when it repeatedly crosses the negative real axis, the phase of $G(u)$ jumps from $-\pi$ to π. Then the phase of $G^\alpha(u)$ changes from $-\alpha\pi$ to $\alpha\pi$, causing a discontinuity when α is not a natural number.

Albrecher et al. [2] found that this happens when the principal value of the complex square root d is selected, as most numerical implementations of complex functions do, but can be avoided if the second value is used instead. Albrecher et al. and Lord and Kahl [31] proved that this alternative representation, originally proposed by Schoutens et al. [39, Eq. (17)], is continuous and gives numerically stable prices in the full-dimensional and unrestricted parameter space:

$$\begin{aligned}\phi(\boldsymbol{\theta}; u, \tau) = \exp\Bigg\{ & iu \log \frac{F}{S_0} \\ & + \frac{\kappa\bar{v}}{\sigma^2}\left[(\xi - d)\tau - 2\log\frac{1 - g_2 e^{-d\tau}}{1 - g_2}\right] + \frac{v_0}{\sigma^2}(\xi - d)\frac{1 - e^{-d\tau}}{1 - g_2 e^{-d\tau}}\Bigg\},\end{aligned} \tag{12}$$

where

$$g_2 := \frac{\xi - d}{\xi + d} = \frac{1}{g_1}. \tag{13}$$

Another equivalent form of the characteristic function was proposed later by del Baño Rollin et al. [15, Eq. (6)]. We correct the expression in that paper by adding the term $-\kappa\bar{v}\rho\tau i u/\sigma$ to the exponent, resulting in

$$\phi(\boldsymbol{\theta}; u, \tau) = \exp\left(iu \log \frac{F}{S_0} - \frac{\kappa\bar{v}\rho\tau i u}{\sigma} - A\right) B^{2\kappa\bar{v}/\sigma^2}, \tag{14}$$

where

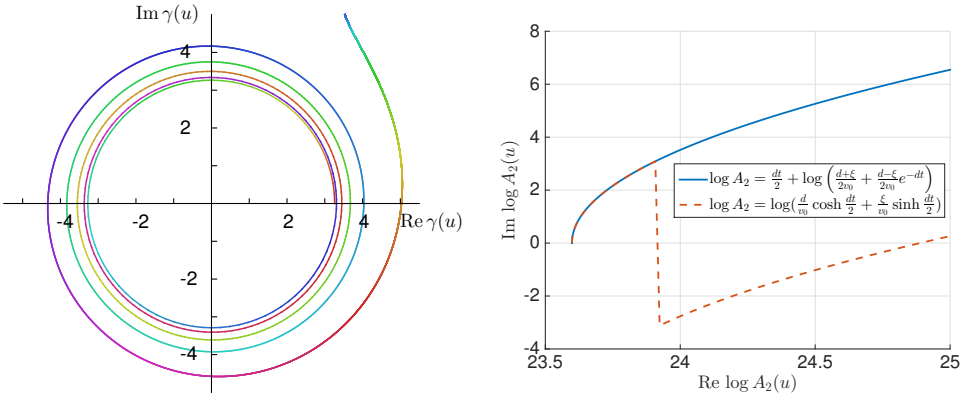
$$A := \frac{A_1}{A_2}, \tag{15a}$$

$$A_1 := (u^2 + iu)\sinh\frac{d\tau}{2}, \tag{15b}$$

$$A_2 := \frac{d}{v_0}\cosh\frac{d\tau}{2} + \frac{\xi}{v_0}\sinh\frac{d\tau}{2}, \tag{15c}$$

$$B := \frac{de^{\kappa\tau/2}}{v_0 A_2}. \tag{15d}$$

Del Baño Rollin et al. introduced their expression to analyse the log-spot density, and since then it has not been used for any other purpose. It was obtained by manipulating the complex moment generating function; besides being more compact, it replaces the exponential functions in the exponent with hyperbolic functions, which makes the derivatives easier. Therefore, we will use this expression to obtain the analytical gradient.



(a) $\gamma(u),\ u \in [0, 500]$. (b) $\log A_2(u),\ u \in [0, 4]$.

Fig. 1: Trajectories of $\gamma(u)$ and two equivalent forms of $\log A_2(u)$ in the complex plane. The curves were generated using the parameters in Table 1 with time to maturity $\tau = 15$.

Table 1: Parameters specification.

Model parameters		Market parameters	
κ	3.00	S_0	1.00
$\bar{v}$	0.10	K	1.10
σ	0.25	r	0.02
ρ	−0.80	q	0
v_0	0.08		

However, the same discontinuity problem pointed out by Kahl and Jäckel appears here too. It comes from the factor $B^{2\kappa\bar{v}/\sigma^2}$, or more specifically from A_2 in the denominator of B. Fig. 1a shows a trajectory of $\gamma(u) := (A_2(u)\log\log|A_2(u)|)/|A_2(u)|$. The double-logarithmic scaling of the radius compensates the rapid outward movement of the spiralling trajectory of $A_2(u)$ [2, 30]. For the curve we adopt the same hue $h \in [0, 1)$ as Kahl and

Jäckel [30], $h := \log_{10}(u+1) \mod 1$, which means that segments of slowly varying colour represent rapid movements of $A_2(u)$ as a function of u.

We thus modify the representation by rearranging $\log A_2$ to

$$\log A_2 = \log\left(\frac{d}{v_0}\cosh\frac{d\tau}{2} + \frac{\xi}{v_0}\sinh\frac{d\tau}{2}\right) \tag{16a}$$

$$= \log\frac{d(e^{d\tau/2}+e^{-d\tau/2}) + \xi(e^{d\tau/2}-e^{-d\tau/2})}{2v_0} \tag{16b}$$

$$= \log\frac{(d+\xi)e^{d\tau/2} + (d-\xi)e^{-d\tau/2}}{2v_0} \tag{16c}$$

$$= \log\left[e^{d\tau/2}\left(\frac{d+\xi}{2v_0} + \frac{d-\xi}{2v_0}e^{-d\tau}\right)\right] \tag{16d}$$

$$= \frac{d\tau}{2} + \log\left(\frac{d+\xi}{2v_0} + \frac{d-\xi}{2v_0}e^{-d\tau}\right). \tag{16e}$$

Fig. 1b shows the trajectories of the two equivalent formulations of $\log A_2$. The rearrangement (16e) resolves the discontinuities arising from the logarithm with Eq. (15c) as an argument. Then we insert Eq. (16e) into $\log B$ and denote the final expression as D:

$$\log B = \log\frac{d}{v_0} + \frac{\kappa\tau}{2} - \log A_2 \tag{17a}$$

$$= \log\frac{d}{v_0} + \frac{(\kappa-d)\tau}{2} - \log\left(\frac{d+\xi}{2v_0} + \frac{d-\xi}{2v_0}e^{-d\tau}\right) =: D. \tag{17b}$$

So we propose a new representation of the characteristic function which is algebraically equivalent to all the previous expressions and does not show the discontinuities of Eqs. (10) and (14) for large maturities:

$$\phi(\boldsymbol{\theta}; u, \tau) = \exp\left(iu\log\frac{F}{S_0} - \frac{\kappa\bar{v}\rho\tau iu}{\sigma} - A + \frac{2\kappa\bar{v}}{\sigma^2}D\right). \tag{18}$$

We have discussed four equivalent representations of the Heston characteristic function, three from previous research and one newly proposed here by us. We compare them in Fig. 2: the plot of our expression is continuous and overlaps Schoutens et al.'s, while the other two exhibit discontinuities due to the multivalued complex functions. Moreover our expression, like the one by del Baño Rollin et al. from which it was obtained, has the advantage of being easily differentiable, as shown in the next section. These properties are summarised in Table 2.

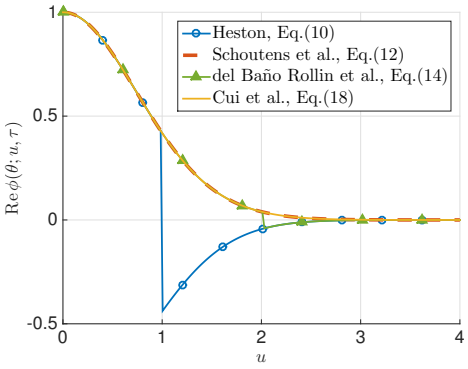


Fig. 2: Four equivalent representations of the Heston characteristic function. The curves were generated using the parameters in Table 1 with time to maturity $\tau = 15$. Eq. (10) jumps at $u = 1$, Eq. (14) jumps at $u = 2$, while Eqs. (12) and (18) are continuous.

Table 2: Properties of the four representations of the Heston characteristic function.

	Numerically continuous	Easily differentiable
Heston	✗	✗
Schoutens et al.	✓	✗
del Baño Rollin et al.	✗	✓
Cui et al.	✓	✓

3.3. Analytical gradient

We use $\boldsymbol{\nabla} = \partial/\partial\boldsymbol{\theta}$ for the gradient operator with respect to the parameter vector $\boldsymbol{\theta}$ and $\boldsymbol{\nabla}\boldsymbol{\nabla}^\intercal$ for the Hessian operator. For convenience, we omit to write the dependence of the residual vector $\boldsymbol{r}$ on $\boldsymbol{\theta}$.

3.3.1. The basic theorem of the analytical gradient

Let $\boldsymbol{J} = \boldsymbol{\nabla}\boldsymbol{r}^\intercal \in \mathbb{R}^{m\times n}$ be the Jacobian matrix of the residual vector $\boldsymbol{r}$ with elements

$$J_{ji} = \left[\frac{\partial r_i}{\partial \theta_j}\right] = \left[\frac{\partial C(\boldsymbol{\theta}; K_i, \tau_i)}{\partial \theta_j}\right], \tag{19}$$

and $\boldsymbol{H}(r_i) := \boldsymbol{\nabla}\boldsymbol{\nabla}^\intercal r_i \in \mathbb{R}^{m\times m}$ be the Hessian matrix of each residual r_i with elements

$$H_{jk}(r_i) = \left[\frac{\partial^2 r_i}{\partial\theta_j \partial\theta_k}\right]. \tag{20}$$

Following the nonlinear least-squares formulation (4)–(5), one can easily write the gradient and Hessian of the objective function f as

$$\boldsymbol{\nabla} f = \boldsymbol{J}\boldsymbol{r}, \tag{21a}$$

$$\boldsymbol{\nabla}\boldsymbol{\nabla}^\intercal f = \boldsymbol{J}\boldsymbol{J}^\intercal + \sum_{i=1}^{n} r_i \boldsymbol{H}(r_i). \tag{21b}$$

Theorem 1. *Assume that an underlying asset S follows the Heston process* (1)*. Let $\boldsymbol{\theta} := [v_0, \bar{v}, \rho, \kappa, \sigma]^\intercal$ be the parameters in the Heston model, $C(\boldsymbol{\theta}; K, \tau)$ be the price of a vanilla call option on S with strike K and time to maturity τ. Then the gradient of $C(\boldsymbol{\theta}; K, \tau)$ with respect to $\boldsymbol{\theta}$ is*

$$\boldsymbol{\nabla} C(\boldsymbol{\theta}; K, \tau) = \frac{e^{-r\tau}}{\pi} \left[\int_0^\infty \operatorname{Re}\left(\frac{e^{-iu \log \frac{K}{S_0}}}{iu} \boldsymbol{\nabla}\phi(\boldsymbol{\theta}; u - i, \tau) \right) \mathrm{d}u \right.$$
$$\left. - K \int_0^\infty \operatorname{Re}\left(\frac{e^{-iu \log \frac{K}{S_0}}}{iu} \boldsymbol{\nabla}\phi(\boldsymbol{\theta}; u, \tau) \right) \mathrm{d}u \right], \tag{22}$$

where $\boldsymbol{\nabla}\phi(\boldsymbol{\theta}; u, \tau) = \phi(\boldsymbol{\theta}; u, \tau)\boldsymbol{h}(u)$, $\boldsymbol{h}(u) := [h_1(u), h_2(u), \ldots, h_5(u)]^\intercal$ with elements

$$h_1(u) = -\frac{A}{v_0}, \tag{23a}$$
$$h_2(u) = \frac{2\kappa}{\sigma^2} D - \frac{\kappa\rho\tau i u}{\sigma}, \tag{23b}$$
$$h_3(u) = -\frac{\partial A}{\partial \rho} + \frac{2\kappa\bar{v}}{\sigma^2 d}\left(\frac{\partial d}{\partial \rho} - \frac{d}{A_2}\frac{\partial A_2}{\partial \rho} \right) - \frac{\kappa\bar{v}\tau i u}{\sigma}, \tag{23c}$$
$$h_4(u) = \frac{1}{\sigma i u}\frac{\partial A}{\partial \rho} + \frac{2\bar{v}}{\sigma^2} D + \frac{2\kappa\bar{v}}{\sigma^2 B}\frac{\partial B}{\partial \kappa} - \frac{\bar{v}\rho\tau i u}{\sigma}, \tag{23d}$$
$$h_5(u) = -\frac{\partial A}{\partial \sigma} - \frac{4\kappa\bar{v}}{\sigma^3} D + \frac{2\kappa\bar{v}}{\sigma^2 d}\left(\frac{\partial d}{\partial \sigma} - \frac{d}{A_2}\frac{\partial A_2}{\partial \sigma} \right) + \frac{\kappa\bar{v}\rho\tau i u}{\sigma^2}; \tag{23e}$$

$\xi, d, A, A_1, A_2, B, D, \phi(\boldsymbol{\theta}; u, \tau)$ are defined in Eqs. (11a)*,* (11b)*,* (15)*,* (17b) *and* (18)*, respectively.*

Proof. Eq. (22) results from differentiating the vanilla option pricing function (9) under the integral sign; this can be done because the integrand and its gradient are continuous. Then the problem reduces to the derivation of the gradient of the characteristic function $\phi(\boldsymbol{\theta}; u, \tau)$. Starting from Eq. (14) and following the chain rule, one can get $\boldsymbol{\nabla}\phi(\boldsymbol{\theta}; u, \tau)$ as discussed below.

Since v_0 and $\bar{v}$ are only in the exponent and are not involved with the definition of A or B, we directly obtain

$$\frac{\partial \phi(\boldsymbol{\theta}; u, \tau)}{\partial v_0} = -\frac{A}{v_0}\phi(\boldsymbol{\theta}; u, \tau), \tag{24}$$
$$\frac{\partial \phi(\boldsymbol{\theta}; u, \tau)}{\partial \bar{v}} = \frac{2\kappa \log B \phi(\boldsymbol{\theta}; u, \tau)}{\sigma^2}. \tag{25}$$

Next we derive the partial derivative with respect to ρ, since it provides some terms that can be reused for the rest. We have

$$\frac{\partial \phi(\boldsymbol{\theta}; u, \tau)}{\partial \rho} = \phi(\boldsymbol{\theta}; u, \tau)\left(-\frac{\kappa\bar{v}\tau i u}{\sigma} - \frac{\partial A}{\partial \rho} \right) + \phi(\boldsymbol{\theta}; u, \tau)\frac{2\kappa\bar{v}}{\sigma^2}\frac{1}{B}\frac{\partial B}{\partial \rho} \tag{26a}$$

$$= \phi(\boldsymbol{\theta}; u, \tau) \left[-\frac{\kappa\bar{v}\tau iu}{\sigma} - \frac{\partial A}{\partial \rho} + \frac{2\kappa\bar{v}}{\sigma^2 d} \left(\frac{\partial d}{\partial \rho} - \frac{d}{A_2} \frac{\partial A_2}{\partial \rho} \right) \right] \tag{26b}$$

$$= \phi(\boldsymbol{\theta}; u, \tau) \left[-\frac{\partial A}{\partial \rho} + \frac{2\kappa\bar{v}}{\sigma^2 d} \left(\frac{\partial d}{\partial \rho} - \frac{d}{A_2} \frac{\partial A_2}{\partial \rho} \right) - \frac{\kappa\bar{v}\tau iu}{\sigma} \right], \tag{26c}$$

where

$$\frac{\partial d}{\partial \rho} = -\frac{\xi\sigma iu}{d}, \tag{27a}$$

$$\frac{\partial A_2}{\partial \rho} = -\frac{\sigma iu(2+\xi\tau)}{2dv_0} \left(\xi \cosh \frac{d\tau}{2} + d \sinh \frac{d\tau}{2} \right), \tag{27b}$$

$$\frac{\partial B}{\partial \rho} = \frac{e^{\kappa\tau/2}}{v_0} \left(\frac{1}{A_2} \frac{\partial d}{\partial \rho} - \frac{d}{A_2^2} \frac{\partial A_2}{\partial \rho} \right), \tag{27c}$$

$$\frac{\partial A_1}{\partial \rho} = -\frac{iu(u^2+iu)\tau\xi\sigma}{2d} \cosh \frac{d\tau}{2}, \tag{27d}$$

$$\frac{\partial A}{\partial \rho} = \frac{1}{A_2} \frac{\partial A_1}{\partial \rho} - \frac{A}{A_2} \frac{\partial A_2}{\partial \rho}. \tag{27e}$$

By merging and rearranging terms, we find that

$$\frac{\partial A}{\partial \kappa} = \frac{i}{\sigma u} \frac{\partial A}{\partial \rho}, \tag{28a}$$

$$\frac{\partial B}{\partial \kappa} = \frac{i}{\sigma u} \frac{\partial B}{\partial \rho} + \frac{B\tau}{2}, \tag{28b}$$

which are inserted into

$$\frac{\partial \phi(\boldsymbol{\theta}; u, \tau)}{\partial \kappa} = \phi(\boldsymbol{\theta}; u, \tau) \left(-\frac{\partial A}{\partial \kappa} + \frac{2\bar{v}}{\sigma^2} \log B + \frac{2\kappa\bar{v}}{\sigma^2 B} \frac{\partial B}{\partial \kappa} - \frac{\bar{v}\rho\tau iu}{\sigma} \right) \tag{29}$$

to reach the expression (23d). Similarly, Eq. (23e) can be obtained by applying the chain rule to Eq. (14), and the intermediate terms for $\partial\phi(\boldsymbol{\theta}; u, \tau)/\partial\sigma$ can be written in terms of those for $\partial\phi(\boldsymbol{\theta}; u, \tau)/\partial\rho$, that is

$$\frac{\partial d}{\partial \sigma} = \left(\frac{\rho}{\sigma} - \frac{1}{\xi} \right) \frac{\partial d}{\partial \rho} + \frac{\sigma u^2}{d}, \tag{30a}$$

$$\frac{\partial A_1}{\partial \sigma} = \frac{(u^2+iu)\tau}{2} \frac{\partial d}{\partial \sigma} \cosh \frac{d\tau}{2}, \tag{30b}$$

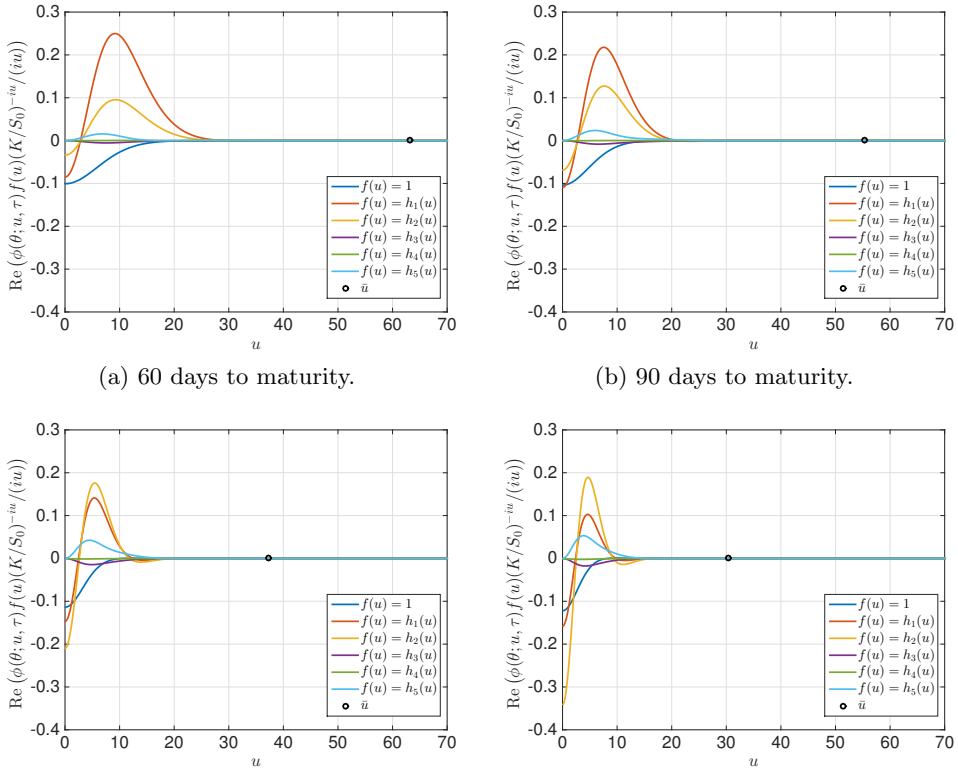
$$\frac{\partial A_2}{\partial \sigma} = \frac{\rho}{\sigma} \frac{\partial A_2}{\partial \rho} - \frac{2+\tau\xi}{v_0\tau\xi iu} \frac{\partial A_1}{\partial \rho} + \frac{\sigma\tau A_1}{2v_0}, \tag{30c}$$

$$\frac{\partial A}{\partial \sigma} = \frac{1}{A_2} \frac{\partial A_1}{\partial \sigma} - \frac{A}{A_2} \frac{\partial A_2}{\partial \sigma}. \tag{30d}$$

In the end, we replace $\log B$ appearing in Eqs. (25) and (29) with D, defined in Eq. (17b), to ensure the numerical continuity of the implementation. □

Next we discuss the computation of the integrands in Eq. (22) and their convergence.

3.3.2. Efficient calculation and convergence of the integrands



(a) 60 days to maturity.

(b) 90 days to maturity.

(c) 180 days to maturity.

(d) 252 days to maturity.

Fig. 3: Convergence of the integrands: $\mathrm{Re}\left(\phi(\boldsymbol{\theta};u,\tau)(K/S_0)^{-iu}/(iu)\right)$ in the Heston pricing formula for $C(\boldsymbol{\theta};K,\tau)$ (dark blue) and $\mathrm{Re}\left(\phi(\boldsymbol{\theta};u,\tau)\boldsymbol{h}(u)(K/S_0)^{-iu}/(iu)\right)$ in the components of its gradient $\partial C/\partial\boldsymbol{\theta}$ (other colours); $h_j(u), j=1,\ldots,5$ are respectively relevant for $\partial C/\partial\theta_j$. The black circle indicates the value $\bar{u}$ where all integrands are below 10^{-8}.

All integrands have the form $\mathrm{Re}\left(\phi(\boldsymbol{\theta};u,\tau)h_j(u)(K/S_0)^{-iu}/(iu)\right)$ and $h_j(u)$ is a product of elementary functions depending on which parameter is under consideration. It has been pointed out in the original paper by Heston [26] that the term $\mathrm{Re}\left(\phi(\boldsymbol{\theta};u,\tau)(K/S_0)^{-iu}/(iu)\right)$ is a smooth function that decays rapidly and presents no difficulties; its product with elementary functions decreases fast too. A visual example is shown in Fig. 3, with parameters given in Table 1. In our time units, $\tau=1$ is a trading year made of 252 days.

Denote as $\bar{u}$ the value of u for which all integrands are not larger than 10^{-8}. For our testing parameter set, we observe in Figs. 3 and 4 that $\bar{u}$ decreases when τ increases. This is due to the fact that the more spread-out a function is, the more localised its Fourier transform is (see the uncertainty principle in physics): as τ increases, the probability density of S_T stretches out, while its Fourier transform $\phi(\boldsymbol{\theta};u,\tau)$ squeezes. More specifically, if X

and U are random variables whose probability density functions are, apart of a constant, Fourier pairs of each other, the product of their variances is a constant, i.e., $\mathrm{Var}(X)\mathrm{Var}(U) \geq 1$. Based on this observation, one can adjust the truncation according to the maturity of the option and hence do fewer integrand evaluations for options with longer maturities.

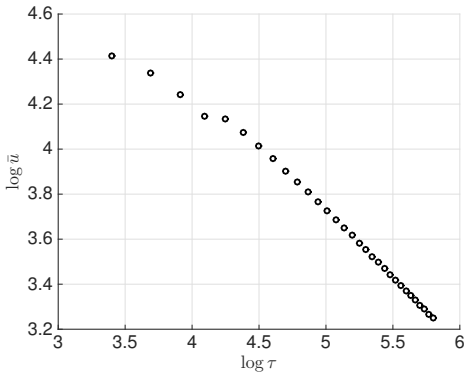


Fig. 4: As the time to maturity τ increases, the value $\bar{u}$ for which all integrands evaluate to 10^{-8} or less decreases.

In order to obtain the integrands in Eq. (22), one only needs to compute $\phi(\boldsymbol{\theta}; u, \tau)$ and $\boldsymbol{h}(u)$. After rearranging and merging terms, we find that calculating $\boldsymbol{h}(u)$ can be boiled down to obtaining the intermediate terms (27), (28) and (30). It is favourable that the components of $\boldsymbol{h}(u)$ share these common terms because so the gradient $\boldsymbol{\nabla} C(\boldsymbol{\theta}; K, \tau)$ can be obtained by vectorising the quadrature for all the integrands as illustrated in Algorithm 3.1.

ALGORITHM 3.1. Vectorised integration in the Heston gradient.

---

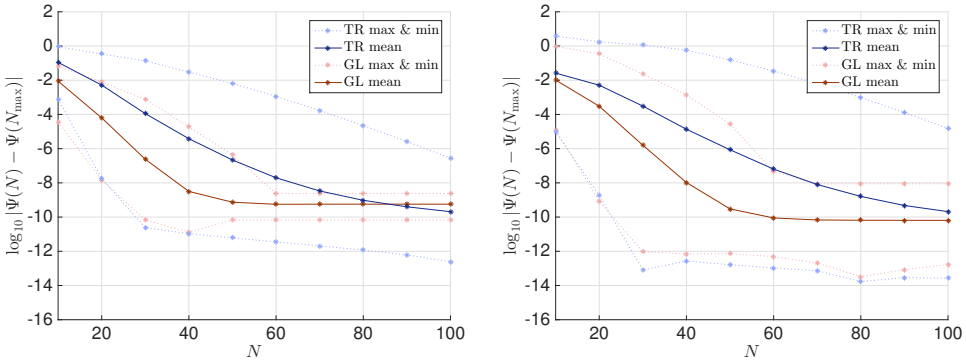
1 Specify N grid nodes $(u_k)_{k=1}^N$ and N corresponding weights $(w_k)_{k=1}^N$.
2 **for** $k = 1, 2, \ldots, N$ **do**
3 Compute $\phi(\boldsymbol{\theta}; u_k, \tau)$ and $\boldsymbol{h}(u_k)$.
4 **end**
5 **for** $j = 1, 2, \ldots, 5$ **do**
6 Compute
$\int_0^\infty \frac{K^{-iu}}{iu} \phi(\boldsymbol{\theta}; u, \tau) h_j(u) \mathrm{d}u \approx \sum_{k=1}^N \frac{K^{-iu_k}}{iu_k} \phi(\boldsymbol{\theta}; u_k, \tau) h_j(u_k) w_k$.
7 **end**

Due to the interdependence among the components of $\boldsymbol{h}(u)$, this scheme is faster than computing and integrating each component $h_j(u)$ individually. Next, we discuss the choice of the numerical integration method and of the key parameters N, u_k and w_k, but we point out that this vectorised quadrature is compatible with any numerical integration method. Also note that the evaluation of $\phi(\boldsymbol{\theta}; u_k, \tau) h_j(u_k)$ is independent of K; therefore for a

given τ one can factorise the computation of $\phi(\boldsymbol{\theta}; u_k, \tau)h_j(u_k)$ across strikes: options on a smile with fixed tenor can share this term, saving computations. We did not implement this, but it may be useful in other applications.

3.3.3. Integration scheme

The computation of the integrals in the pricing function (9) and the gradient function (22) dominates the cost of calibration. Thus, we discuss the proper choice of the numerical integration scheme. Specifically, we compare the trapezoidal rule (TR) and the Gauss-Legendre rule (GL). The integration error for the pricing formula and its gradient is plotted in Figs. 5a and 5b. The horizontal axis reports the number of quadrature nodes N and the



(a) for the pricing function (b) for the gradient function

Fig. 5: Integration error in the Heston model: comparison between TR (full blue line for the mean and dotted purple line for the maximum and minimum) and GL (full red line for the mean and dotted pink line for the maximum and minimum).

vertical axis the decadic logarithm of the absolute error of integration. The latter is defined as

$$\varepsilon_{\text{integration}} := |\Phi(N) - \Phi(N_{\max})|, \tag{31}$$

where $\Phi(N)$ is the value of the integration with N nodes, N is selected equidistantly in the range $[10, 100]$, $N_{\max}$ should be ∞ and is chosen as 1000 in our case. For the plots we use 40 options with different strikes and maturities. More details on these options are given in Section 5.

The error converges faster for GL than TR and has always a smaller variation when more options are involved. In order to achieve an average accuracy of 10^{-8}, GL requires 40 nodes and TR requires 70. In order for the integrations for all the options to achieve an accuracy at 10^{-8}, GL requires 60 nodes and TR requires much more than 100.

Besides a fast convergence of its integration error, GL has an advantageous selection of nodes. GL rescales the domain of integration to $[-1, +1]$, selects nodes that are symmetric around the origin, and assigns the same

weight to each symmetric pair of nodes. Thus, a further reduction in computation can be achieved by making use of the common terms of a node and its opposite. Based on these benefits, we choose the GL integration scheme with about 60 nodes to calibrate the Heston model.

3.3.4. Comparison with numerical gradient

Previous calibration methods approximate the gradient by a finite difference scheme. A central difference scheme is the approximation

$$\boldsymbol{\nabla} C(\boldsymbol{\theta}; K, \tau) \approx \frac{C(\boldsymbol{\theta} + \boldsymbol{\epsilon}; K, \tau) - C(\boldsymbol{\theta} - \boldsymbol{\epsilon}; K, \tau)}{2\epsilon}, \tag{32}$$

where $\boldsymbol{\epsilon} := \epsilon \boldsymbol{e}$ and ϵ is small. Different values of the increment ϵ could be chosen for each component θ_j; for simplicity we have taken it constant. The size of the difference, ϵ, has a non-trivial effect on the approximation. An excessively small value of ϵ is not able to reflect the overall function behavior at the point and may lead to a wrong moving direction. Moreover the numerical gradient naturally has an error and one cannot expect to find a solution with a better accuracy than that of the gradient. In most cases, the iteration stagnates when the error of the objective function is roughly the same size as the error of the gradient.

Besides the instability caused by an inappropriate choice of ϵ, a numerical gradient has a higher computational cost than an analytical gradient. Recall that the evaluation of one option price $C(\boldsymbol{\theta}; K, \tau)$ requires the evaluation of two integrals as in Eq. (9). Let n be the number of options to be calibrated. At each iteration, one needs to compute $20n$ integrals if using the finite difference scheme while only $2n$ integrals if using the analytical form with the vectorised integration scheme. To give a more intuitive comparison between the two methods, we perform a preliminary experiment with $\epsilon = 10^{-4}$ and $n = 40$ using the MATLAB function `quadv` with an adaptive Simpson rule for the numerical integration. In Table 3, we report the CPU time as an average of 500 runs and the number of calls of the integral function for each method. In order to give a relative sense of speed that is independent of the machine, the CPU time for analytical gradient is scaled to unity, and that for numerical gradient results about 16 times longer.

Table 3: A comparison between numerical and analytical gradients for $n = 40$ options.

Computational cost	Numerical gradient	Analytical gradient
CPU time (arbitrary units)	15.8	1.0
Number of integral evaluations	800	80

Considering the 94% of saving in computational time and the exempt from deciding ϵ, we propose to use the analytical Heston gradient with vec-

torised quadrature in a gradient-based optimisation algorithm to calibrate the model.

4. Calibration using the Levenberg-Marquardt method

In this section, we present the algorithm for a complete and fast calibration of the Heston model using the LM method [36]. The LM method is a typical tool to solve a nonlinear least-squares problem like Eq. (4). The search step is given by

$$\Delta\boldsymbol{\theta} = (\boldsymbol{J}\boldsymbol{J}^{\intercal} + \mu\boldsymbol{I})^{-1}\boldsymbol{\nabla} f, \tag{33}$$

where $\boldsymbol{I}$ is the identity matrix and μ is a damping factor. By adaptively adjusting μ, the method changes between the steepest descent method and the Gauss-Newton method: when the iterate is far from the optimum, μ is given a large value so that the Hessian matrix is dominated by the scaled identity matrix

$$\boldsymbol{\nabla}\boldsymbol{\nabla}^{\intercal} f \approx \mu\boldsymbol{I}; \tag{34}$$

when the iterate is close to the optimum, μ is assigned a small value so that the Hessian matrix is dominated by the Gauss-Newton approximation

$$\boldsymbol{\nabla}\boldsymbol{\nabla}^{\intercal} f \approx \boldsymbol{J}\boldsymbol{J}^{\intercal}, \tag{35}$$

which omits the second term $\sum_{i=1}^{n} r_i \boldsymbol{H}(r_i)$ in Eq. (21b). The approximation (35) is reliable when either r_i or $\boldsymbol{H}(r_i)$ is small. The former happens when the problem is a so-called *small residual problem* and the latter happens when f is nearly linear. The viewpoint is that the model should yield small residuals around the optimum because otherwise it is an inappropriate model. The Heston model has been known to be able to explain the smile and skew of the volatility surface. Therefore, we conjecture it to be a small residual problem and adopt the approximation of the Hessian in Eq. (35) as converging to the optimum. There are various implementations of the LM method, such as `MINPACK` [16], `LEVMAR` [32], `sparseLM` [33] etc. We adopt the `LEVMAR` package which is a robust and stable implementation in C/C++ distributed under GNU. Although its documentation does not report a use in computational finance, `LEVMAR` has been integrated into many open source and commercial products in other applications such as astrometric calibration and image processing. See Algorithm 4.1.

In lines 1 and 5 of Algorithm 4.1, the option pricing function is evaluated. In lines 1 and 8, the gradient function is evaluated. In line 2, the parameters ω and ν_0 are given the default values of `LEVMAR`. In line 4, a 5×5 linear system is solved; in `LEVMAR` this is done by an LDLT factorisation with the pivoting strategy of Bunch and Kaufman [9] using the `LAPACK` [4] routine.

ALGORITHM 4.1. Levenberg-Marquardt algorithm to calibrate the Heston model.

1 Given the initial guess $\boldsymbol{\theta}_0$, compute $\|\boldsymbol{r}(\boldsymbol{\theta}_0)\|$ and $\boldsymbol{J}_0$.
2 Choose the initial damping factor $\mu_0 = \omega \max\{\text{diag}(\boldsymbol{J}_0)\}$ and $\nu_0 = 2$.
3 **for** $k = 0, 1, 2, \ldots$ **do**
4 Solve the normal equations (33) for $\Delta\boldsymbol{\theta}_k$.
5 Compute $\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k + \Delta\boldsymbol{\theta}_k$ and $\|\boldsymbol{r}(\boldsymbol{\theta}_{k+1})\|$.
6 Compute $\delta_L = \Delta\boldsymbol{\theta}_k{}^\intercal(\mu_k\Delta\boldsymbol{\theta}_k + \boldsymbol{J}_k\boldsymbol{r}(\boldsymbol{\theta}_k))$ and $\delta_F = \|\boldsymbol{r}(\boldsymbol{\theta}_k)\| - \|\boldsymbol{r}(\boldsymbol{\theta}_{k+1})\|$.
7 **if** $\delta_L > 0$ *and* $\delta_F > 0$ **then**
8 Accept the step: compute $\boldsymbol{J}_{k+1}$, $\mu_{k+1} = \mu_k$, $\nu_{k+1} = \nu_k$.
9 **else**
10 Recalculate the step: set $\mu_k = \mu_k\nu_k$, $\nu_k = 2\nu_k$ and repeat from line 4.
11 **end**
12 **if** *the stopping criterion* (36) *is met* **then**
13 Break.
14 **end**
15 **end**

The stopping criterion for the LM algorithm is when one of the following is satisfied:

$$\|\boldsymbol{r}(\boldsymbol{\theta}_k)\| \leq \varepsilon_1, \tag{36a}$$

$$\|\boldsymbol{J}_k\boldsymbol{e}\|_\infty \leq \varepsilon_2, \tag{36b}$$

$$\frac{\|\Delta\boldsymbol{\theta}_k\|}{\|\boldsymbol{\theta}_k\|} \leq \varepsilon_3, \tag{36c}$$

where $\varepsilon_1, \varepsilon_2$ and ε_3 are tolerance levels. The first condition (36a) indicates that the iteration is stopped by a desired value of the objective function (4) and (5). The second condition (36b) indicates that the iteration is stopped by a small gradient. The third condition (36c) indicates that the iteration is stopped by a stagnating update.

5. Numerical results

In this section, we present our experimental results for the calibration of the Heston model. We first describe our synthetic data and then report the performance of our calibration method in comparison with the fastest previous method. We examine the Hessian matrix at an optimal solution, which reveals the reason of the multiple optima observed in previous research. In the end, we test on three parameterisations that are typical for certain options. The result justifies the computational efficiency and robustness of our method for practical problems.

5.1. Data

To check whether the optimal parameter set found by the algorithm is the global optimum, we used the parameter set $\boldsymbol{\theta}^*$ specified in Table 1 to generate the implied volatility surface in Table 4. Its 40 values are characterised by the call options with $\Delta^{\text{call}} = 50\%,\ 25\%$ and 10%, and by the put options with $\Delta^{\text{put}} = -25\%$ and -10%, each with maturities from 30 to 360 days. As we assume $q = 0$, $\Delta^{\text{call}} = 50\%$ corresponds to the standard ATM delta-neutral straddle; moreover, the put-call parity $C - P = e^{-r\tau}(F - K)$ implies $\Delta^{\text{call}} - \Delta^{\text{put}} = e^{-q\tau} = 1$, i.e. $\Delta^{\text{put}} = -25\%$ and -10% correspond to $\Delta^{\text{call}} = 75\%$ and 90%. Here Δ^{call} and Δ^{put} are the BS delta of a call or put option, i.e. the sensitivity of the BS call price C_{BS} or the BS put price P_{BS} with respect to the underlying spot price S_t: to obtain the relevant strikes, one solves for K the equations $\Delta^{\text{call}}_{\text{BS}} := \partial C_{\text{BS}}(K,\tau)/\partial S_t = 0.50,\ 0.25$ and 0.10, and the equations $\Delta^{\text{put}}_{\text{BS}} := \partial P_{\text{BS}}(K,\tau)/\partial S_t = -0.10$ and -0.25. The FX convention of quoting option strikes by their delta has the effect that values close in the strike axis have a comparable likelihood, which is not the case with strikes quoted as absolute numbers. The target is thus to find a

Table 4: Synthetic implied volatility surface $\sigma_{\text{BS}}(K,\tau)$ used for calibration. The implied volatility σ_{BS} matches the BS price to the Heston price, i.e. $C_{\text{BS}}(\sigma_{\text{BS}}; K, \tau) = C(\boldsymbol{\theta}^*; K, \tau)$, with the parameter set $\boldsymbol{\theta}^*$ given in Table 1. The strikes K are computed from the BS deltas reported in the header.

τ (days)	$\Delta^{\text{put}}_{\text{BS}} = -10\%$	$\Delta^{\text{put}}_{\text{BS}} = -25\%$	$\Delta^{\text{call}}_{\text{BS}} = 50\%$	$\Delta^{\text{call}}_{\text{BS}} = 25\%$	$\Delta^{\text{call}}_{\text{BS}} = 10\%$
30	2.5096	1.4359	0.2808	0.2540	0.2369
60	2.4351	1.3216	0.2847	0.2606	0.2417
90	2.3823	1.2955	0.2878	0.2660	0.2489
120	2.3383	1.2677	0.2904	0.2699	0.2548
150	2.2996	1.2407	0.2925	0.2745	0.2598
180	2.2619	1.2166	0.2943	0.2777	0.2641
252	2.1767	1.1671	0.2975	0.2837	0.2722
360	2.0618	1.1136	0.3007	0.2897	0.2803

parameter set $\boldsymbol{\theta}^\dagger$ that can replicate the volatility surface in Table 4. If $\boldsymbol{\theta}^\dagger$ is far from $\boldsymbol{\theta}^*$ or, in other words, depends on the initial guess $\boldsymbol{\theta}_0$, then one concludes that local optimal parameter sets exist. Otherwise the problem presents only a global optimum.

We validated our method using different optimal parameters and initial guesses within a reasonable range given in Table 5. The procedure is described in Algorithm 5.1. The Feller condition $2\kappa\bar{v}/\sigma^2 > 1$, which ensures that the volatility $v(t)$ is always strictly larger than zero rather than just non-negative ($2\kappa\bar{v}/\sigma^2$ is known as the Feller ratio), was not enforced by us

either generating the parameters and guesses or as a constraint in the LM method because prices are nevertheless legitimate and the Feller condition is often violated in practice [2], especially in the FX market [13, Table 6.3]. Recently it has been observed that the Feller condition is probably the most counterfactual assumption in finance, and relaxing it is not only harmless from the modelling point of view, but also effective for derivative pricing, because its violation is a natural consequence of the misspecification of affine models [37].

Table 5: Reasonable ranges to randomly generate Heston model parameters and the average absolute distance between the initial guess $\boldsymbol{\theta}_0$ and the optimum $\boldsymbol{\theta}^*$.

Range for model parameters		Absolute deviation from $\boldsymbol{\theta}^*$	
κ	$(0.50, 5.00)$	$\lvert\kappa_0 - \kappa^*\rvert$	1.5097
$\bar{v}$	$(0.05, 0.95)$	$\lvert\bar{v}_0 - \bar{v}^*\rvert$	0.2889
σ	$(0.05, 0.95)$	$\lvert\sigma_0 - \sigma^*\rvert$	0.2875
ρ	$(-0.90, -0.10)$	$\lvert\rho_0 - \rho^*\rvert$	0.2557
v_0	$(0.05, 0.95)$	$\lvert(v_0)_0 - v_0^*\rvert$	0.3063

ALGORITHM 5.1. Validation procedure.

for $i = 1, 2, \ldots, 100$ **do**
  Generate a vector of parameters $\boldsymbol{\theta}_i^*$, each component of which is an independent uniformly distributed random number in the interval specified in Table 5.
  for $j = 1, 2, \ldots, 100$ **do**
    Generate an initial guess $\boldsymbol{\theta}_{0j}$, each component of which is an independent uniformly distributed random number in the interval specified in Table 5.
    Validate Algorithm 4.1 using the initial guess $\boldsymbol{\theta}_{0j}$ to find $\boldsymbol{\theta}_i^*$.
  end
end

Following this procedure, we validated Algorithm 4.1 with 10 000 test cases. An average of the distances between the initial guesses $\boldsymbol{\theta}_0$ and the optima $\boldsymbol{\theta}^*$ is given in Table 5. The results of the tests are discussed in the next section.

5.2. Performance

The computations were performed on a MacBook Pro with a 2.6 GHz Intel Core i5 processor, 8 GB of RAM and OS X Yosemite version 10.10.5. The pricing and gradient functions for the Heston model were coded in C++

using Xcode version 7.3.1. We used `LEVMAR` version 2.6 [32] as the LM solver setting the tolerances in Eqs. (36) to $\varepsilon_1 = \varepsilon_2 = \varepsilon_3 = 10^{-10}$. However, in our experiments the LM iteration was always stopped by meeting the condition on the objective function (36a). We used GL integration with $N = 64$ nodes and for simplicity we truncated the upper limit of the integration in Eq. (22) at $\bar{u} = 200$ which in all cases is enough for pricing and calibrating. The code is provided in the supplementary material.

The proposed method succeeded in finding the presumed parameter set in 9 843 cases out of 10 000 without any constraints on the search space and in 9 856 cases restraining the search to the intervals specified in Table 5. The average CPU time for the whole calibration process was less than 0.3 seconds. See Table 6 for detailed information on the whole validation set. In Table 7 and in the rest of this section we specify the information for a representative example with the parameters $\boldsymbol{\theta}^*$ specified in Table 1 and the initial guess $\boldsymbol{\theta}_0 = [1.20, 0.20, 0.30, -0.60, 0.20]^\intercal$.

Table 6: Optimisation: average results of 10 000 test cases.

Absolute deviation from $\boldsymbol{\theta}^*$		Error measure		Computational cost	
$\lvert\kappa^\dagger - \kappa^*\rvert$	1.54×10^{-3}	$\lVert\boldsymbol{r}_0\rVert$	1.39×10^{-1}	CPU time (seconds)	0.29
$\lvert\bar{v}^\dagger - \bar{v}^*\rvert$	2.40×10^{-5}	$\lVert\boldsymbol{r}^\dagger\rVert$	2.94×10^{-11}	LM iterations	12.82
$\lvert\sigma^\dagger - \sigma^*\rvert$	3.79×10^{-3}	$\lVert\boldsymbol{J}^\dagger\boldsymbol{e}\rVert_\infty$	1.47×10^{-5}	Price evaluations	14.57
$\lvert\rho^\dagger - \rho^*\rvert$	1.52×10^{-2}	$\lVert\Delta\boldsymbol{\theta}^\dagger\rVert$	3.21×10^{-4}	Gradient evaluations	12.82
$\lvert v_0^\dagger - v_0^*\rvert$	6.98×10^{-6}			Linear systems solved	13.57

Table 7: Optimisation of the example specified in Table 1.

Absolute deviation from $\boldsymbol{\theta}^*$		Error measure		Computational cost	
$\lvert\kappa^\dagger - \kappa^*\rvert$	1.09×10^{-3}	$\lVert\boldsymbol{r}_0\rVert$	4.73×10^{-2}	CPU time (seconds)	0.29
$\lvert\bar{v}^\dagger - \bar{v}^*\rvert$	2.18×10^{-6}	$\lVert\boldsymbol{r}^\dagger\rVert$	1.00×10^{-12}	LM iterations	13
$\lvert\sigma^\dagger - \sigma^*\rvert$	4.70×10^{-5}	$\lVert\boldsymbol{J}^\dagger\boldsymbol{e}\rVert_\infty$	1.21×10^{-5}	price evaluations	14
$\lvert\rho^\dagger - \rho^*\rvert$	9.89×10^{-6}	$\lVert\Delta\boldsymbol{\theta}^\dagger\rVert$	2.50×10^{-4}	gradient evaluations	13
$\lvert v_0^\dagger - v_0^*\rvert$	1.18×10^{-6}			linear systems solved	13

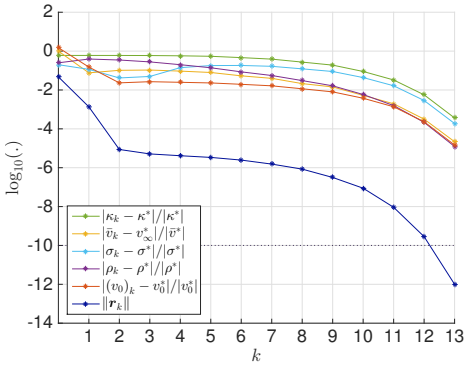
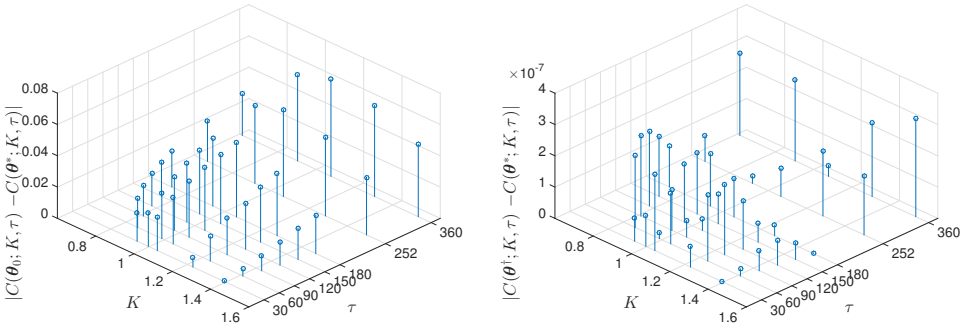


Fig. 6: Convergence of the LM method.

The convergence of the residual $\boldsymbol{r}_k$ and the relative distance of each parameter towards the optimum is plotted in Fig. 6. In Figs. 7a and 7b, we plot the pricing error on the implied volatility surface at the initial point $\boldsymbol{\theta}_0$ and the optimal point $\boldsymbol{\theta}^\dagger$, respectively. As can be seen, the pricing error decreases from 10^{-2} to 10^{-7} after 13 steps.



(a) at the initial point $\boldsymbol{\theta}^0$ (b) at the final point $\boldsymbol{\theta}^\dagger$

Fig. 7: Pricing error on the implied volatility surface.

This result contrasts the conclusion of previous research: local optimal parameters are not intrinsically embedded in the Heston calibration problem, but rather caused by an objective function shaped as a narrow valley with a flat bottom and a premature stopping criterion.

We plot the contours for $\|\boldsymbol{r}\|$ when varying 2 out of 5 parameters. Starting from $\boldsymbol{\theta}_0$, the iteration path is shown with contour plots in Fig. 8. The initial point $\boldsymbol{\theta}_0$ is marked with a black circle and the true solution $\boldsymbol{\theta}^*$ is marked with a black plus symbol. The red lines with asterisks are the iteration paths of $\boldsymbol{\theta}_k$, $k = 1, \ldots, 13$. For almost all pairs, the first step is a long steepest descent step that is nearly orthogonal to the contour. The rest are relatively cautious steps with the Gauss-Newton approximation of the Hessian. The contour plots do not show evidence for local minima, at least

not in 2-dimensional sections.

The Gauss-Newton approximation of the Hessian matrix at the optimal solution is given in Table 8. The Hessian matrix is ill-conditioned with a

Table 8: Hessian matrix $\boldsymbol{\nabla}\boldsymbol{\nabla}^{\intercal} f(\boldsymbol{\theta}^*)$ at the optimum of the example specified in Table 1.

	$\partial\kappa$	$\partial\bar{v}$	$\partial\sigma$	$\partial\rho$	∂v_0
$\partial\kappa$	5.26×10^{-5}				
$\partial\bar{v}$	9.65×10^{-3}	$2.26 \times 10^{+1}$			
$\partial\sigma$	-5.49×10^{-4}	-7.66×10^{-2}	7.46×10^{-3}		
$\partial\rho$	1.61×10^{-4}	2.00×10^{-2}	-2.34×10^{-3}	7.56×10^{-4}	
∂v_0	5.28×10^{-3}	$1.18 \times 10^{+1}$	-3.53×10^{-2}	8.40×10^{-3}	9.69×10^{-1}

condition number of 3.98×10^6. The elements $\partial^2 f(\boldsymbol{\theta}^*)/\partial\kappa^2$ and $\partial^2 f(\boldsymbol{\theta}^*)/\partial\rho^2$ are of a much smaller order than the others. This suggests that the objective function, when around the optimum, is less sensitive to changes of κ and ρ. The effect of κ on the objective function is weak because option prices depend on the integrated volatility, which is little sensitive to the degree of oscillation of volatility; ρ controls the slope of the smile, so this parameter is difficult to identify if a narrow range of moneyness is used for calibration.

In other words, the objective function is more stretched along these two axes as can be verified looking at the contours, for example in Figs. 5.2 and 5.2. The ratio between $\partial^2 f(\boldsymbol{\theta}^*)/\partial\kappa^2$ and $\partial^2 f(\boldsymbol{\theta}^*)/\partial\bar{v}^2$ is of order 10^{-6}, which indicates a great disparity in sensitivity: changing 1 unit of $\bar{v}$ is comparable to changing 10^6 units of κ. On the other hand, this explains the so-called local minima reported in previous research. When one starts from a different initial point and stops the iteration with a high tolerance, it is possible that the iterate lands somewhere in the region where κ and ρ are very different.

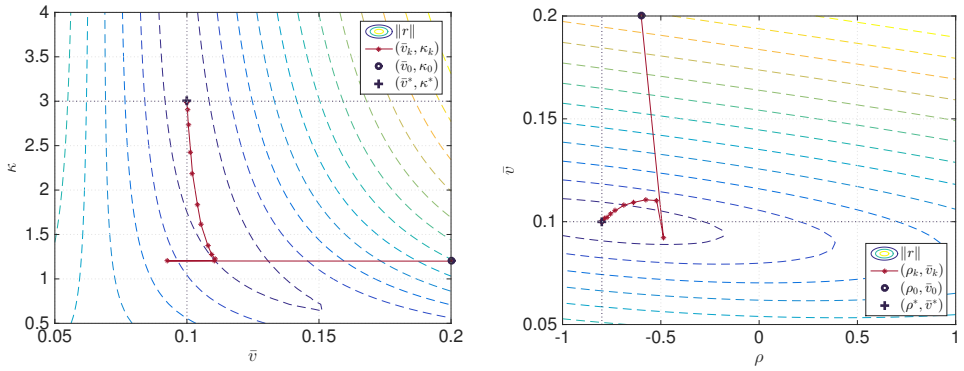


Fig. 8: Contours of $\|\boldsymbol{r}\|$ and iteration path for (θ_i, θ_j).

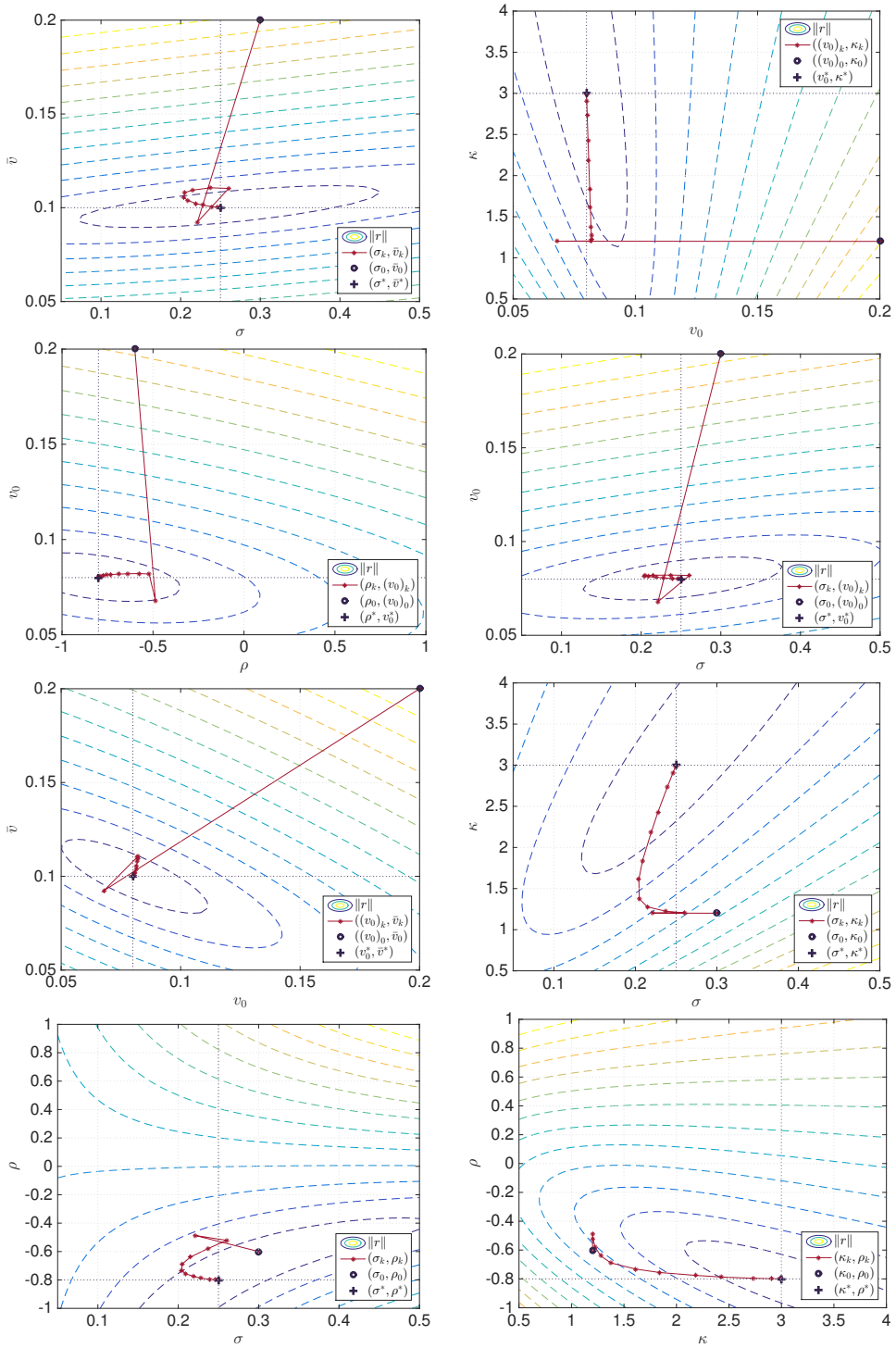


Fig. 8: (continued) Contours of $\|\boldsymbol{r}\|$ and iteration path for (θ_i, θ_j).

There are two possible approaches that one can seek to deal with this: the first is to scale the parameters to a similar order and search on a better-scaled objective function; the second is to decrease the tolerance level for the optimisation process, meaning to approach the very bottom of this objective function.

In Table 9, we present the performance of the LM method with analytical gradient (LMA), the LM method with numerical gradient (LMN), and a feasibility perturbed sequential quadratic programming method (FPSQP) [22] adopted in UniCredit bank. As the concrete implementation of FPSQP is owned by the bank, we only extract their test results. The computational cost can be compared through the number of evaluations of the pricing function (9) per iteration, expressed as a multiple of the number n of options to be calibrated. LMA requires about n pricing function evaluations per

Table 9: Performance comparison between solvers.

	LMA	LMN	FPSQP
Stopping criterion	$\|\boldsymbol{r}(\boldsymbol{\theta}_k)\| \leq 10^{-10}$	$\|\boldsymbol{r}(\boldsymbol{\theta}_k)\| \leq 10^{-10}$	$\|\boldsymbol{\Delta\theta}_k\| \leq 10^{-6}$
Iterations	13	22	-
Price evaluations per iteration	$1.08n$	$1.70n$	$6.00n$

step. LMN requires more for the gradient approximation, but the difference is not large since LMN uses a rank-one update for the subsequent Jacobian matrices. FPSQP requires about 5.5 times more than LMA and achieves only a lower accuracy for the stopping criterion for the gradient.

We tested our method also on a few realistic model parameterisations. In Table 10, we present three test cases that are representative respectively for long-dated FX options, long-dated interest rate options and equity options [3]. They are believed to be prevalent and challenging for the simulation of Heston model [25]. Each component of the initial guess is an independent

Table 10: Test cases with realistic Heston model parameters. Case I: long-dated FX options. Case II: long-dated interest rate options. Case III: equity options.

	Case I	Case II	Case III
κ^*	0.50	0.30	1.00
$\bar{v}^*$	0.04	0.04	0.09
σ^*	1.00	0.90	1.00
ρ^*	-0.90	-0.50	-0.30
v_0^*	0.04	0.04	0.09

uniformly distributed random number in the ±10% range of the corresponding optimum. This choice is due to the fact that practitioners usually choose the initial guess as the last available estimation which is expected to be close to the solution if the calibration is frequent enough and the market does not change drastically. We test each case with 100 initial guesses. Our previous test range in Table 5 has covered these cases too, but here we would like to focus on the performance of our method when applied to these typical examples and thus justify its computational efficiency and robustness for practical application. The information about the convergence as an average of the 100 initial guesses is given in Table 11. For the practical cases with initial guesses in the vicinity, it takes less than or around one second to obtain the optimal solution.

Table 11: Calibration results for three typical realistic cases, reporting an average on 100 initial guesses for each of them.

		Case I	Case II	Case III
Absolute deviation from $\boldsymbol{\theta}^*$	$\lvert\kappa^\dagger - \kappa^*\rvert$	2.87×10^{-2}	1.35×10^{-3}	1.20×10^{-3}
	$\lvert\bar{v}^\dagger - \bar{v}^*\rvert$	4.80×10^{-3}	4.52×10^{-5}	2.11×10^{-5}
	$\lvert\sigma^\dagger - \sigma^*\rvert$	5.29×10^{-2}	7.48×10^{-4}	3.94×10^{-4}
	$\lvert\rho^\dagger - \rho^*\rvert$	3.65×10^{-2}	1.69×10^{-5}	1.46×10^{-5}
	$\lvert v_0^\dagger - v_0^*\rvert$	2.14×10^{-3}	1.46×10^{-5}	1.07×10^{-5}
Error measure	$\lVert\boldsymbol{r}_0\rVert$	2.70×10^{-4}	4.51×10^{-5}	1.02×10^{-4}
	$\lVert\boldsymbol{r}^\dagger\rVert$	1.12×10^{-4}	9.24×10^{-11}	3.33×10^{-11}
	$\lVert\boldsymbol{J}^\dagger\boldsymbol{e}\rVert_\infty$	1.77×10^{-1}	4.63×10^{-6}	4.15×10^{-6}
	$\lVert\Delta\boldsymbol{\theta}^\dagger\rVert$	6.88×10^{-21}	1.63×10^{-8}	5.10×10^{-5}
Computational cost	CPU time	0.40	1.11	0.15
	LM iterations	16.83	51.52	6.86
	Price evaluations	23.38	52.60	7.86
	Gradient evaluations	16.83	51.52	6.86
	Linear systems solved	23.38	51.52	6.86

6. Conclusion

We proposed a new representation of the Heston characteristic function which is easy to differentiate and avoids discontinuities due to inconsistent choices of the branch of multivalued complex functions. We derived the analytical form of the gradient of the Heston option pricing function with respect to the model parameters. The result can be applied in any gradient-based algorithm. We gave an algorithm for a full and fast calibration of

the Heston model using the Levenberg-Marquardt method, which succeeds in finding the global optimal parameter set within a reasonable number of iterations. We validated our algorithm with randomly generated guesses and parameters as well as three typical cases of Heston model parameterisations for long-dated FX options, long-dated interest rate options and equity options. The resulting parameters replicate the volatility surface with an l_2-norm error of 10^{-10} and an l_1-norm error around 10^{-7}. The speed and stability gained by the use of our analytical gradient make the proposed method suitable for the purpose of live trading and risk management. We discussed several numerical issues. We examined a final Hessian matrix and plotted sample contours of the objective function around the global optimum. To find the latter, we point out that either a rescaling of the parameters or a low tolerance level is needed.

There are still margins of improvement for this calibration algorithm. For example, the integrals in the pricing formula (9) can be consolidated [38, Eq. 1.71], so that only a single numerical integration is required instead of two; this is likely to increase the efficiency and reduce the error, although the former is already high and the latter quite small. Due to the oscillations of the characteristic function, one could test adaptive quadrature methods like Gauss-Kronrod as high-precision benchmarks. It may be interesting to extend the results to include log-normal price jumps; these are usually added to the Heston dynamics to capture the short-term implied volatility smile for maturities of one week and below, which is not well reproduced by continuous stochastic volatility models. It is also worth to investigate how the choice of the objective function and of the observation points affect the calibration performance, and to repeat the numerical tests on real market data.

Acknowledgements

We thank Gianluca Fusai, Giuseppe Di Poto, Gabriele Pompa, Stefano De Marco, Stefano Herzel and Lucio Fiorin for useful comments. The support of the Economic and Social Research Council (ESRC) in funding the Systemic Risk Centre is gratefully acknowledged (Grant number ES/K002309/1).

[1] Aït-Sahalia, Y., R. Kimmel. 2007. Maximum likelihood estimation of stochastic volatility models. *Journal of Financial Economics* **83**(2) 413–452. doi:10.1016/j.jfineco.2005.10.006.

[2] Albrecher, H., P. Mayer, W. Schoutens, J. Tistaert. 2007. The little Heston trap. *Wilmott Magazine* **2007**(January) 83–92. URL `http://www.wilmott.com/pdfs/121218_heston.pdf`.

[3] Andersen, L. 2008. Simple and efficient simulation of the Heston stochastic volatility model. *Journal of Computational Finance* **11**(3) 1–42. doi:10.21314/JCF.2008.189.

[4] Anderson, E., Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, D. Sorensen. 1999. *LAPACK Users' Guide*. 3rd ed. Society for Industrial and Applied Mathematics, Philadelphia, PA. doi:10.1137/1.9780898719604.

[5] Battauz, Anna, Marzia De Donno, Alessandro Sbuelz. 2014. The put-call symmetry for American options in the Heston stochastic volatility model. *Mathematical Finance Letters* **7** 1–8. URL `http://scik.org/index.php/mfl/article/view/1840`.

[6] Bauer, R. 2012. Fast calibration in the Heston model. URL `http://www.fam.tuwien.ac.at/~sgerhold/pub_files/theses/bauer.pdf`. Master's thesis, Vienna University of Technology.

[7] Beliaeva, N., Sanjay K. Nawalkha. 2010. A simple approach to pricing American options under the Heston stochastic volatility model. *Journal of Derivatives* **17**(4) 25–43. doi:10.2139/ssrn.1107934.

[8] Black, F., M. Scholes. 1973. The pricing of options and corporate liabilities. *Journal of Political Economy* **81**(3) 631–659. doi:10.1086/260062.

[9] Bunch, J., L. Kaufman, B. Parlett. 1976. Decomposition of a symmetric matrix. *Numerische Mathematik* **27**(1) 95–109. doi:10.1007/BF01399088.

[10] Chen, B. 2007. Calibration of the Heston model with application in derivative pricing and hedging. URL `http://repository.tudelft.nl/islandora/object/uuid:25dc8109-4b39-44b8-8722-3fd680c7c4ad?collection=education`. Master's thesis, Technical University of Delft.

[11] Christoffersen, Peter, Steven Heston, Kris Jacobs. 2009. The shape and term structure of the index option smirk: Why multifactor stochastic volatility models work so well. *Management Science* **55**(12) 1914–1932. URL `http://www.jstor.org/stable/40539255`.

[12] Christoffersen, Peter, Kris Jacobs. 2004. The importance of the loss function in option valuation. *Journal of Financial Economics* **72**(2) 291–318. doi:10.1016/j.jfineco.2003.02.001.

[13] Clark, I. J. 2011. *Foreign Exchange Option Pricing: A Practitioner's Guide*. Wiley, Chichester. doi:10.1002/9781119208679.ch10.

[14] Cont, Rama, Peter Tankov. 2003. *Financial Modelling with Jump Processes*, *Financial Mathematics Series*, vol. 2. Chapman and Hall/CRC, London. doi:10.1201/9780203485217.

[15] del Baño Rollin, S., A. Ferreiro-Castilla, F. Utzet. 2010. On the density of log-spot in the Heston volatility model. *Stochastic Processes and their Applications* **120** 2037–2063. doi:10.1016/j.spa.2010.06.003.

[16] Devernay, F. 2007. C/C++ MINPACK. URL `http://devernay.free.fr/hacks/cminpack`.

[17] Fatone, L., F. Mariani, M. C. Recchioni, F. Zirilli. 2014. The calibration of some stochastic volatility models used in mathematical finance. *Open Journal of Applied Sciences* **4**(2) 23–33. doi:10.4236/ojapps.2014.42004.

[18] Fernández, J., A. Ferreiro, J. García-Rodríguez, A. Leitao, J. López-Salas, C. Vázquez. 2013. Static and dynamic SABR stochastic volatility models: Calibration and option pricing using GPUs. *Mathematics and Computers in Simulation* **94** 55–75. doi:10.1016/j.matcom.2013.05.007.

[19] Fusai, Gianluca, Guido Germano, Daniele Marazzina. 2016. Spitzer identity, Wiener-Hopf factorization and pricing of discretely monitored exotic options. *European Journal of Operational Research* **251**(1) 124–134. doi:10.1016/j.ejor.2015.11.027.

[20] Fusai, Gianluca, Andrea Roncoroni. 2008. *Implementing Models in Quantitative Finance: Methods and Cases*. Springer-Verlag Berlin, Philadelphia, PA. doi:10.1007/978-3-540-49959-6.

[21] Gatheral, J. 2006. *The Volatility Surface: A Practitioner's Guide*, *Finance*, vol. 357. Wiley, Hoboken, NJ. doi:10.1002/9781119202073.

[22] Gerlich, F., A. M. Giese, J. Maruhn, E. Sachs. 2012. Parameter identification in financial market models with a feasible point SQP algorithm. *Computational Optimization and Applications* **51**(3) 1137–1161. doi:10.1007/s10589-010-9369-8.

[23] Gilli, M., E. Schumann. 2012. Calibrating option pricing models with heuristics. A. Brabazon, M. O'Neill, D. Maringer, eds., *Natural Computing in Computational Finance*, *Studies in Computational Intelligence*, vol. 380. Springer, Berlin, 9–37. doi:10.1007/978-3-642-23336-4_2.

[24] Gilli, M., E. Schumann. 2012. Heuristic optimisation in financial modelling. *Annals of Operations Research* **193**(1) 129–158. doi:10.1007/s10479-011-0862-y.

[25] Glasserman, P., K. Kim. 2011. Gamma expansion of the Heston stochastic volatility model. *Finance and Stochastics* **15**(2) 267–296. doi:10.1007/s00780-009-0115-y.

[26] Heston, S. L. 1993. A closed-form solution for options with stochastic volatility and applications to bond and currency options. *Review of Financial Studies* **6**(2) 327–343. doi:10.1093/rfs/6.2.327.

[27] Hurn, A. S., K. A. Lindsay, A. J. McClelland. 2015. Estimating the parameters of stochastic volatility models using option price data. *Journal of Business and Economic Statistics* **33**(4) 579–594. doi:10.1080/07350015.2014.981634.

[28] Jacquier, A., C. Martini. 2011. Heston 2010. doi:10.2139/ssrn.1769744. SSRN 1769744.

[29] Janek, A., T. Kluge, R. Weron, U. Wystup. 2011. FX smile in the Heston model. P. Cizek, W. Karl Härdle, R. Weron, eds., *Statistical Tools for Finance and Insurance*. Springer, Berlin, 133–162. doi:10.1007/978-3-642-18062-0_4.

[30] Kahl, C., P. Jäckel. 2005. Not-so-complex logarithms in the Heston model. *Wilmott Magazine* **2005**(September) 94–103. URL `http://www.wilmott.com/pdfs/110208_heston.pdf`.

[31] Lord, Roger, Christian Kahl. 2010. Complex logarithms in Heston-like models. *Mathematical Finance* **20**(4) 671–694. doi:10.1111/j.1467-9965.2010.00416.x.

[32] Lourakis, M. I. A. 2004. levmar: Levenberg-Marquardt nonlinear least squares algorithms in C/C++. URL `http://www.ics.forth.gr/~lourakis/levmar`.

[33] Lourakis, M. I. A. 2010. Sparse non-linear least squares optimization for geometric vision. K. Daniilidis, P. Maragos, N. Paragios, eds., *Computer Vision — ECCV 2010*, *Image Processing, Computer Vision, Pattern Recognition, and Graphics*, vol. 6311. Springer, Berlin, 43–56. doi:10.1007/978-3-642-15552-9_4.

[34] Mikhailov, S., U. Nögel. 2003. Heston's stochastic volatility model: Implementation, calibration and some extensions. *Wilmott Magazine* **2003**(July) 74–79. URL `http://www.wilmott.com/pdfs/051111_mikh.pdf`.

[35] Moodley, N. 2009. The Heston model: A practical approach with MATLAB code. URL `http://math.nyu.edu/~atm262/fall06/compmethods/a1/nimalinmoodley.pdf`. Bachelor's thesis, University of the Witwatersrand.

[36] Moré, J. J. 1978. The Levenberg-Marquardt algorithm: implementation and theory. G. A. Watson, ed., *Numerical Analysis*, *Lecture Notes in Mathematics*, vol. 630. Springer, Berlin, 105–116. doi: 10.1007/BFb0067700.

[37] Pacati, Claudio, Gabriele Pompa, Roberto Renò. 2017. It's time to relax (the Feller condition). URL `https://sites.google.com/site/qfw2017`. Working paper, presented at the XVIII Workshop on Quantitative Finance, Milan, Italy, 25–27 January 2017.

[38] Rouah, F. D. 2013. *The Heston Model and its Extensions in Matlab and C#*. Wiley, Hoboken, New Jersey. doi:10.1002/9781118656471.

[39] Schoutens, W., E. Simons, J. Tistaert. 2004. A perfect calibration! Now what? *Wilmott Magazine* **2004**(March) 66–78. URL `http://www.wilmott.com/pdfs/070319_schoutens.pdf`.